Einführung

Vorlesung 1, Maschinelles Lernen

Dozenten: Prof. Dr. M. O. Franz, Prof. Dr. O. Dürr

HTWG Konstanz, Fakultät für Informatik

Übersicht

Übersicht

Programme programmieren sich selbst

Maschinelles Lernen: Oberbegriff für die „künstliche“ bzw. automatische Generierung von Wissen aus Erfahrung. Teilgebiete:

- **Klassifikation** (von lat. *classis*, „Klasse“, und *facere*, „machen“) oder **Mustererkennung:** Zusammenfassen von Objekten zu Klassen. Den Eingangs- größen werden *diskrete* Ausgangsgrößen zugewiesen (d.h. die jeweilige Bezeichnung der Klasse).
- **Regression:** Ähnlich wie Klassifikation, aber hier werden den Inputwerten *kontinuierliche* Ausgangsgrößen zugewiesen, d.h. es wird eine Abbildung vom Raum der Eingangsgrößen in einen Ausgangsgrößenraum erlernt.
- **Dichteschätzung:** Erlernen der Wahrscheinlichkeitsverteilung von Daten
- **Rangfolgenbestimmung (Ranking):** Erlernen einer Abbildung vom Eingangsraum auf ordinale Ausgangsgrößen

Lernen anhand von Beispielen

- **Methode des maschinellen Lernens:** Synthese eines Algorithmus oder Modellszur Lösung eines Problems aus Beispielen
- **Überwachtes Lernen:** Zu jedem Trainingsbeispiel ist die Klassenzugehörigkeit bekannt.
- **Unüberwachtes Lernen:** Klassen müssen über „natürliche Gruppen“ gefunden werden (Clustering).
- **Halbüberwachtes Lernen:** die Klassenzugehörigkeit ist nur für ein paar Trainingsbeispiele bekannt, die anderen müssen sinnvoll eingeordnet werden.
- **Bekräftigungslernen (Reinforcement Learning):** Aktionen des Mustererkenners werden nur mit falsch oder richtig bewertet, die Klassenzugehörigkeit wird nicht mitgeteilt.

Allgemeine Struktur der Daten

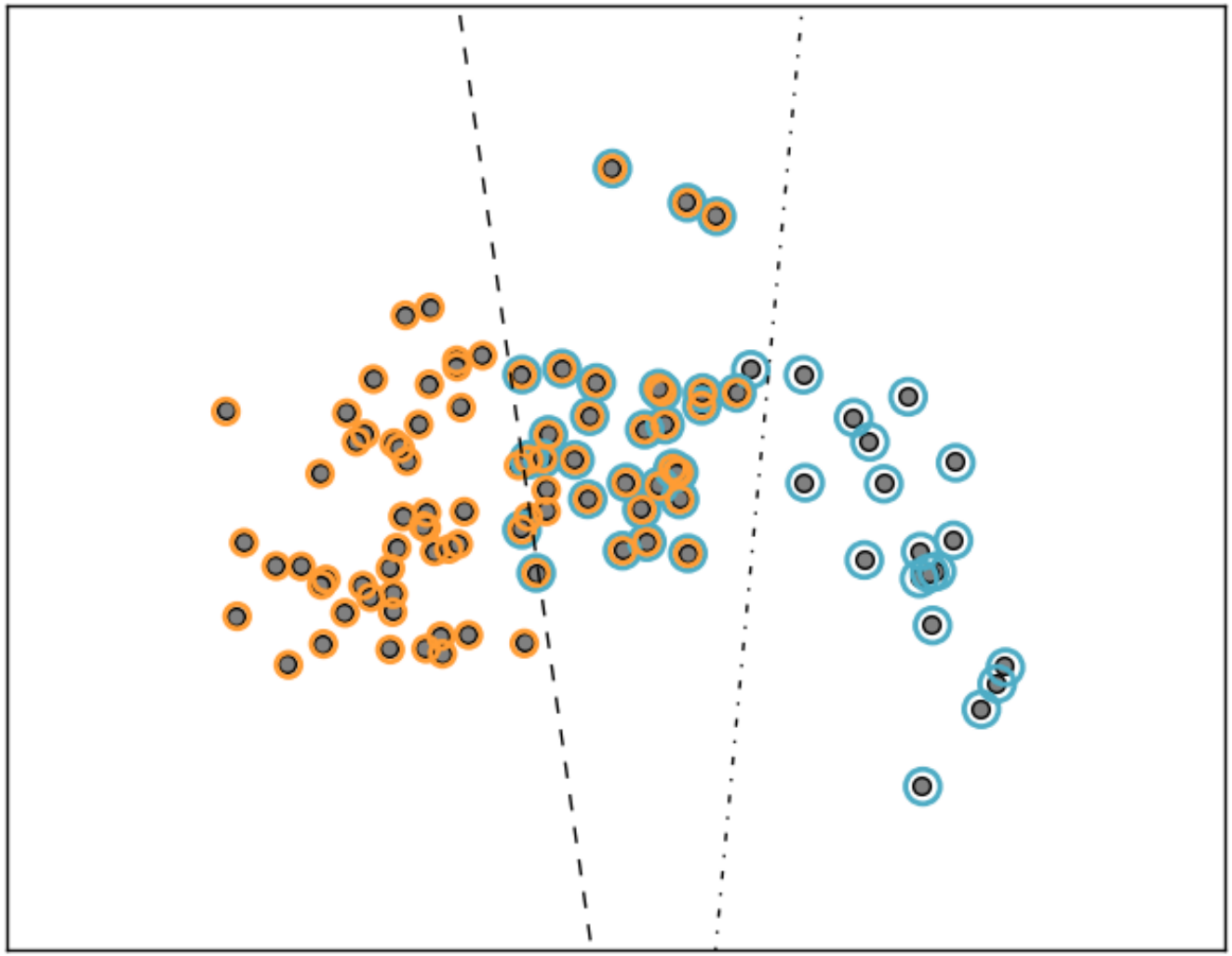
- Die Daten liegen typischerweise in Form von **Objekten** und ihren **Attributen** vor (ähnlich eines Excel-Sheets).
- Ein Attribut (oft auch als Variable oder Feature bezeichnet) ist eine Eigenschaft einen Objekts.
- Eine Sammlung von Attributen beschreibt ein Objekt (oft auch Individuum, Instanz, Fall, Datum, Record oder Beispiel).

p Features

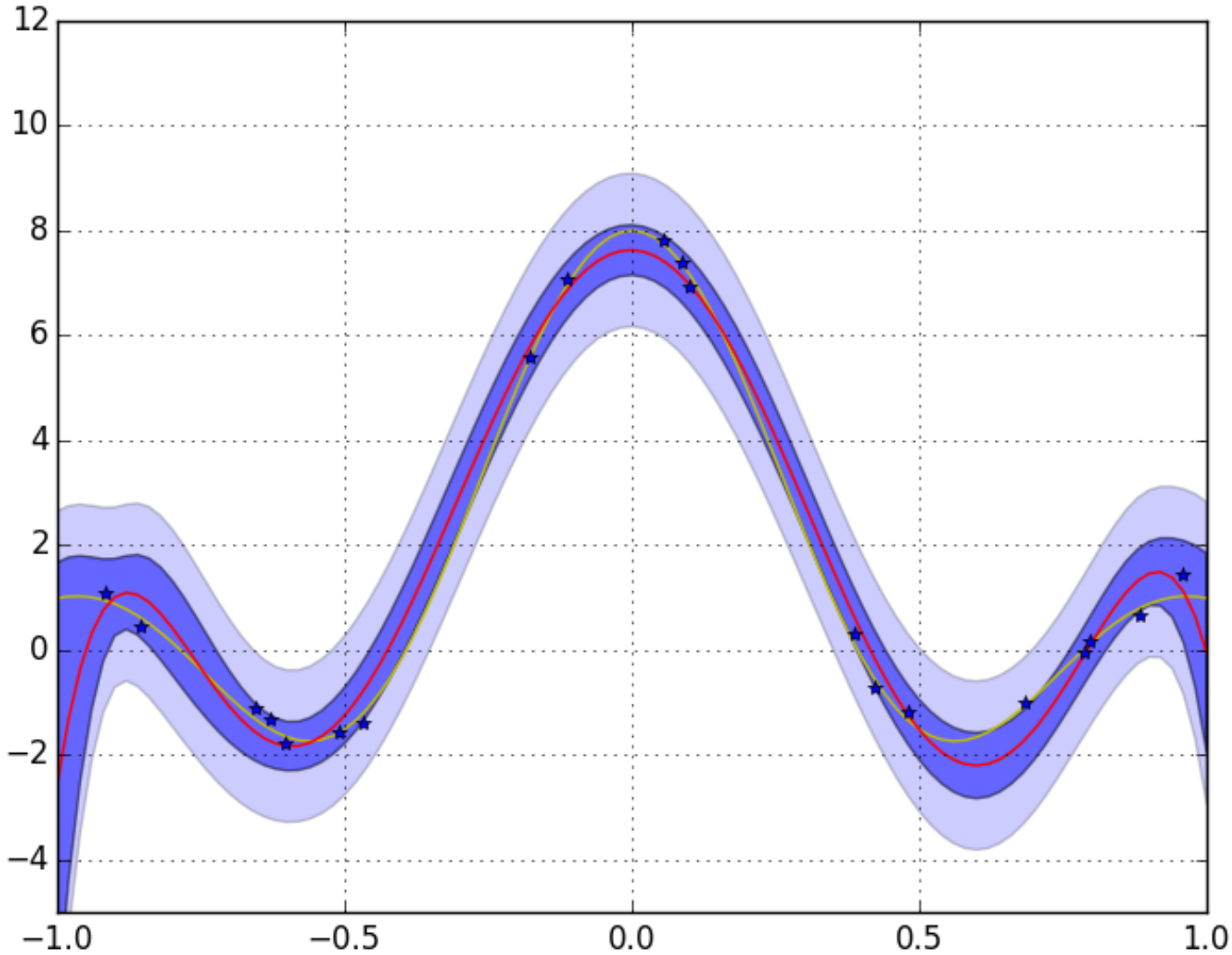
n Objekte

Blume	Type	Sepal.Length	Sepal. Width	Petal.Length	Petal. Width
1	setosa	5.1	3.5	1.4	0.2
2	setosa	4.9	3	1.4	0.2
3	virinica	3.3	3.2	1.6	0.5
4	setosa	5.1	3.5	1.4	0.2
...	...	...	...	...	...
150	virinica	4.9	3	1.4	0.2

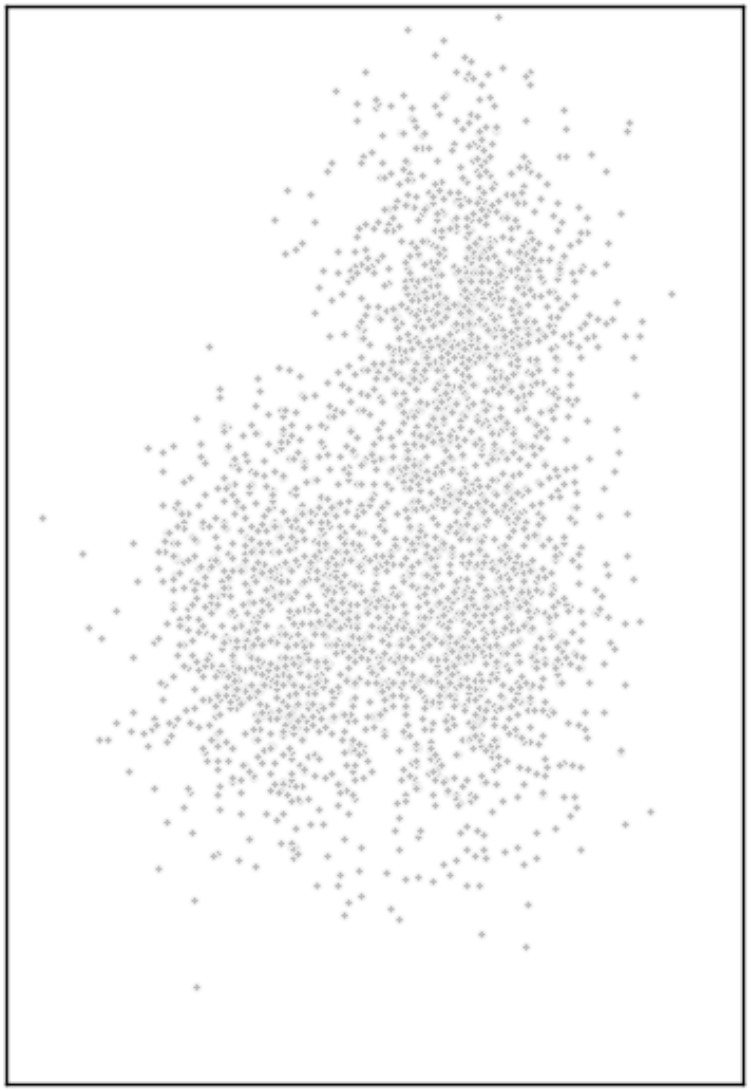
Überwachtes Lernen: Klassifikation

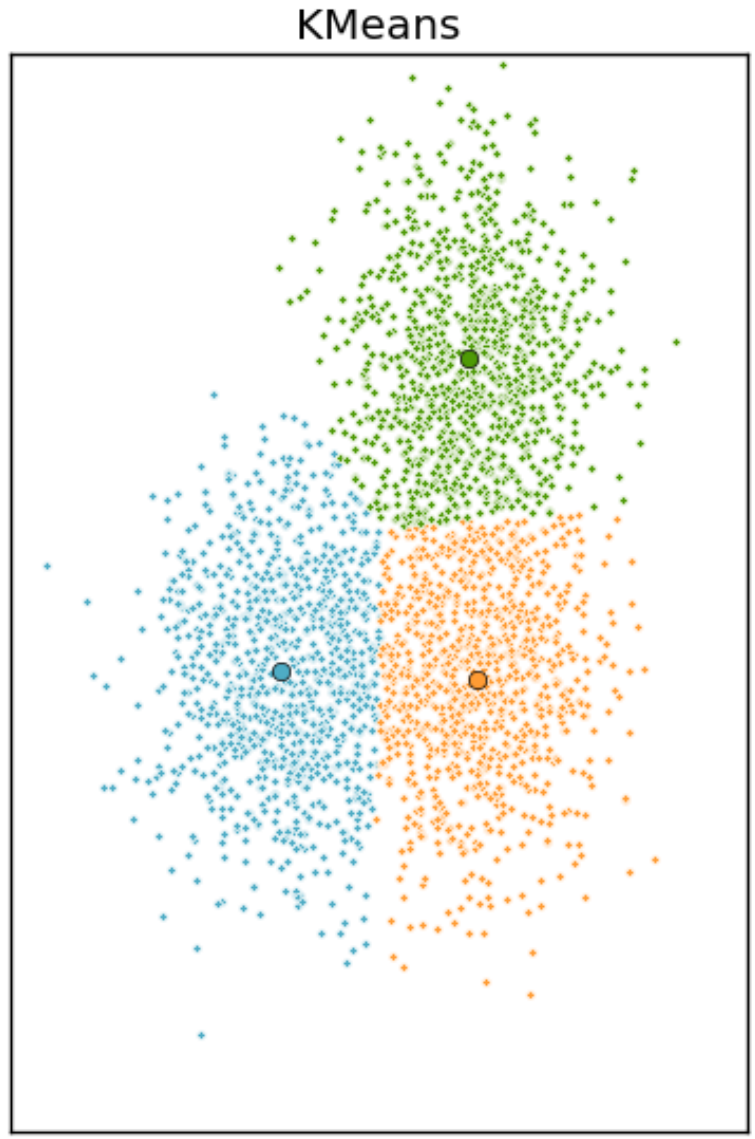


Überwachtes Lernen: Regression

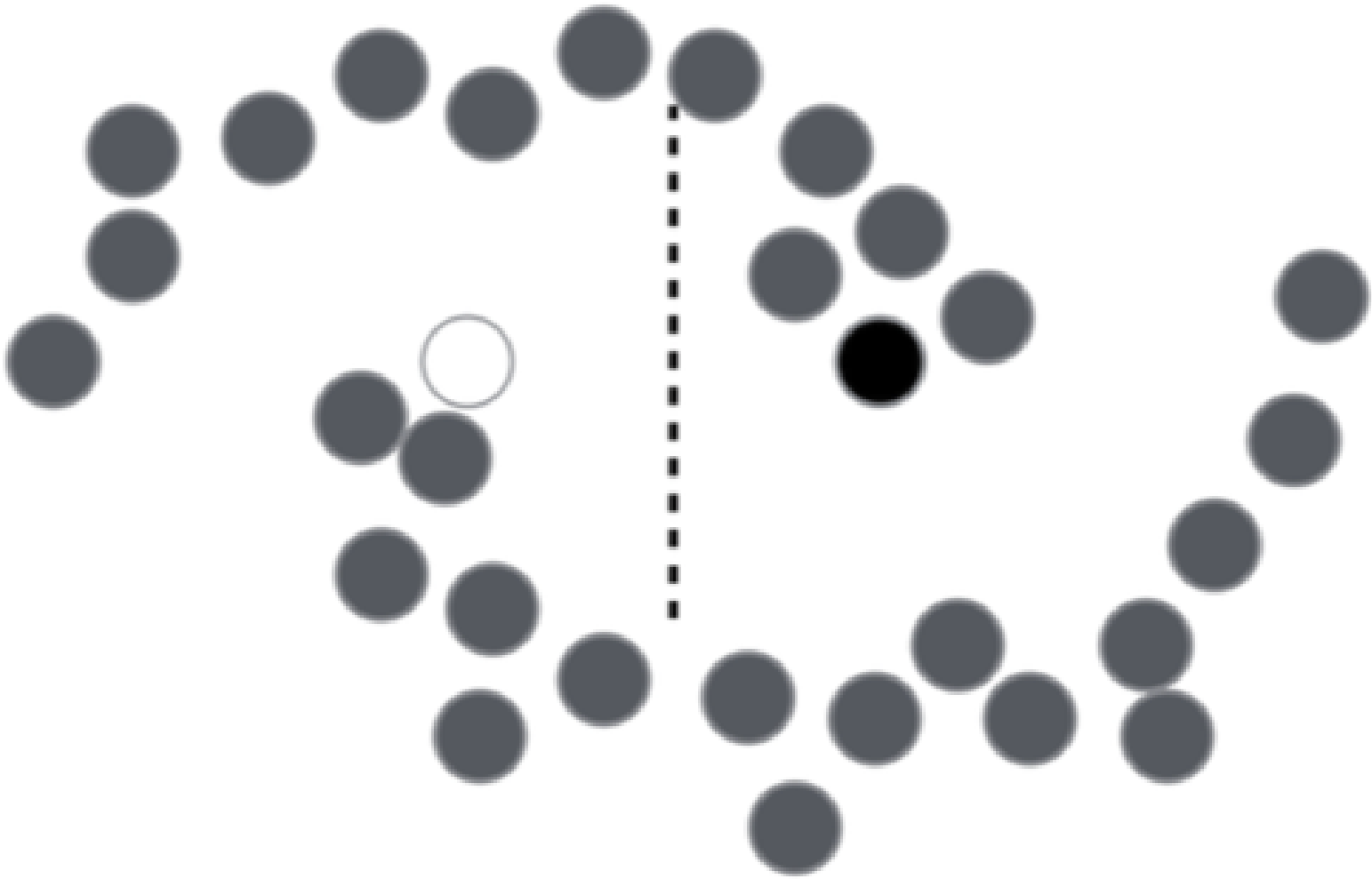


Unüberwachtes Lernen



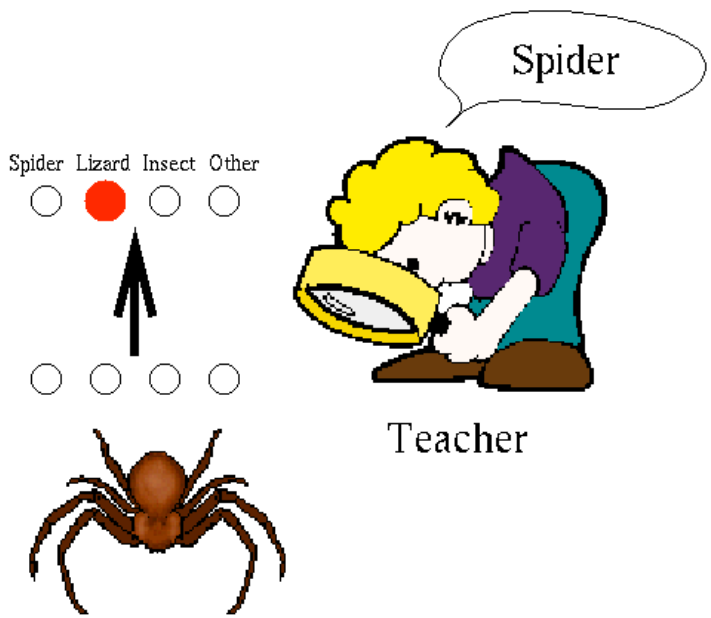


Halbüberwachtes Lernen

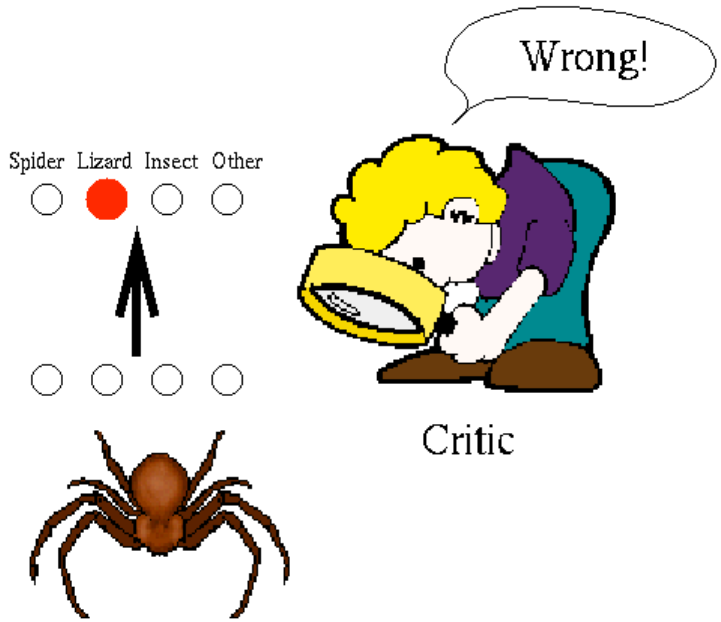


Quelle: TapaniRaiko

Überwachtes und Bekräftigungslernen



Überwachtes Lernen



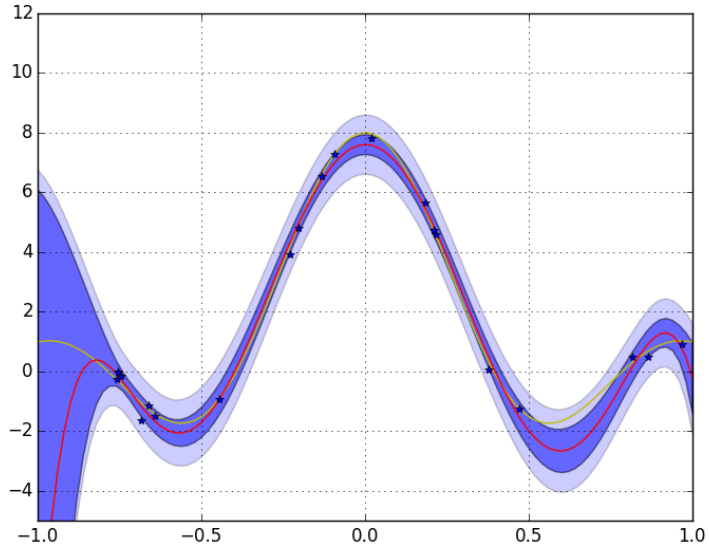
Reinforcement-Lernen

[S. Dennis, 1997]

Grundbegriffe

Lernproblem: Unbekannter Zusammenhang zwischen Ein- und Ausgabewerten (gelb)

Lösung des Lernproblems (Hypothese): Schätzung des Zusammenhangs (rot) aus den Trainingsdaten (Sterne)



Lernalgorithmus oder **Lernmaschine:** Algorithmus, der eine Lösung aus dem Hypothesenraum (hier: Polynome) aufgrund der Trainingsdaten auswählt.

Vorhersage, Prädiktion: Anwendung der Lösung des Lernproblems zur Schätzung des Ausgangswertes zu einem bisher unbekannten Eingangswert.

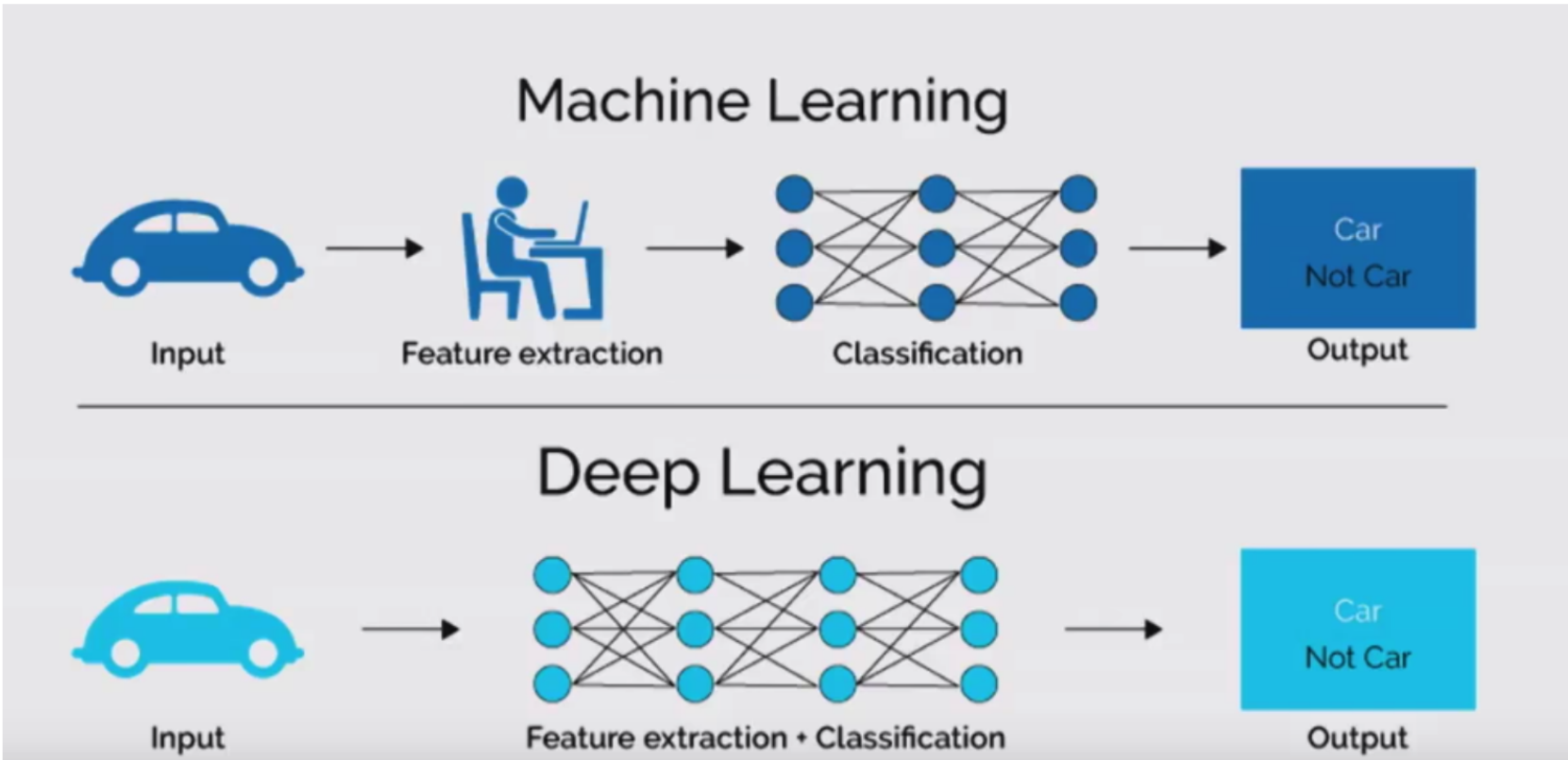
Konsistenz und Generalisierung

- **Konsistenz:** Lösung/Hypothese ist genau an Trainingsdaten angepasst.
- **Generalisierung:** Eigenschaft einer Lösung, auf bisher ungesehenen Eingaben korrekte Vorhersagen/Klassifikationen zu liefern.
- **Overfitting:** Überanpassung, Rauschen der Daten und Fehler werden mitmodelliert.
- **Ockhams Rasiermesser:** wähle immer die einfachste, möglichst konsistente Lösung.

Verschiedene Ansätze im maschinellen Lernen unterscheiden sich durch die jeweilige Gewichtung von Komplexität der Lösung und Genauigkeit auf den Trainingsdaten.

Was ist mit Deep Learning?

Deep Learning ist eine spezielle Form des maschinellen Lernens, bei der die Lernmaschine aus multiplen, hintereinandergeschalteten Verarbeitungsschichten besteht. Die Schichten extrahieren zunehmend komplexere Merkmale aus dem Input. Diese Merkmale werden gelernt, nicht vorgegeben.



Quelle: Siraj Raval

Übersicht

Aufbau der Vorlesung

- Findet in der ersten Semesterhälfte statt.
- 4 Stunden Vorlesung (ergibt 2 SWS)
- 2 Stunden Praktikum (wöchentlich, ergibt 1 SWS)
- Diese Vorlesung muss als Voraussetzung belegt werden, um an der Vorlesung *Deep Learning* in der zweiten Semesterhälfte teilnehmen zu konnen.
- Vorlesung und Praktikum werden mithilfe der Lernplattform Moodle abgehalten. URL: https://moodle.htwg-konstanz.de/moodle/(Standard HTWG-Rechenzentrum-Login)
- Vorlesungsfolien, ergänzendes Material und Praktikumsunterlagen finden sich unter MSI Informatik, Maschinelles Lernen. Bitte manuell einschreiben.

ECTS-Punkte, Arbeitsaufwand und Prüfung

Ziel: Die Grundlagen des maschinellen Lernens kennen lernen; selbstandige Erarbeitung von Teilen des Stoffes in Lekture.

Arbeitsaufwand:

- 5 ECTS-Punkte
- 45 Stunden Kontaktzeit
- 105 Stunden Vor- und Nachbearbeitung

Prüfung:

- Pünktliche Abgabe aller Praktikumsübungen
- 90-minütige Klausur

Überblick (1)

Einführung

- Terminologie des maschinellen Lernens
- Reproduzierbare Forschung
- Aufbereitung und Vorverarbeitung von Daten
- Einfache Methoden der Dimensionsreduktion

Lineare Lernmaschinen

- Perzeptron
- Lineare Regression
- Polynomregression / Kernelized Linear Regression
- Ridge Regression
- Logistische Regression

Überblick (2)

Bayes-Klassifikator

- Entscheidungstheorie
- Bayes-Klassifikator
- Signalentdeckungstheorie
- Naiver Bayes-Klassifikator
- Bias Variance Tradeoff
- Kreuzvalidierung

Supportvektormaschinen

- Kerninduzierte Merkmalsräume
- Klassifikation in hochdimensionalen und nichtlinearen Problemen
- Klassifikation mit maximaler Trennbreite, Supportvektoralgorithmus

Überblick (3)

Kombination von Klassifikatoren

- Entscheidungsbäume
- Bagging
- Random Forests
- Boosting

Literatur

Grundlage der Vorlesung:

- T. Hastie & R. Tibshirani, Introduction to statistical learning, https://www.dataschool.io/15-hours-of-expert-machine-learning-videos/
- R.O.Duda, P.E.Hart & D.G.Stork, Pattern Classification, Wiley, 654 Seiten, 2001.
- N. Cristianini & J. Shawe-Taylor, An introduction to Support Vector Machines, Cambridge University Press, 189 Seiten, 2000.

Weiterführende Literatur:

- T. Hastie, R. Tibshirani & J. Friedman, The elements of statistical learning, Springer, 533 Seiten, 2001
- C.M.Bishop, Pattern recognition and machine learning; Springer, 738 Seiten, 2005.
- B. Schölkopf & A. Smola, Learning with kernels, MIT Press, 644 Seiten, 2002

Übersicht

The Excel Spreadsheet Error… ("The student who caught out the profs")

Reinhardt & Rogoff (2010): Wichtiges Grundlagenpaper für Austeritätspolitik

"... By typing AVERAGE(L30:L44) at one point instead of AVERAGE(L30:L49), they left out Belgium, a key counterexample..."

Brad Plumer, The Washington Post 2013

	B	C	I	J	K	L	M
2			Real GDP growth				
3			Debt/GDP				
4	Country	Coverage	30 or less	30 to 60	60 to 90	90 or above	30 or less
26			3.7	3.0	3.5	1.7	5.5
27	Minimum		1.6	0.3	1.3	-1.8	0.8
28	Maximum		5.4	4.9	10.2	3.6	13.3
29							
30	US	1946-2009	n.a.	3.4	3.3	-2.0	n.a.
31	UK	1946-2009	n.a.	2.4	2.5	2.4	n.a.
32	Sweden	1946-2009	3.6	2.9	2.7	n.a.	6.3
33	Spain	1946-2009	1.5	3.4	4.2	n.a.	9.9
34	Portugal	1952-2009	4.8	2.5	0.3	n.a.	7.9
35	New Zealand	1948-2009	2.5	2.9	3.9	-7.9	2.6
36	Netherlands	1956-2009	4.1	2.7	1.1	n.a.	6.4
37	Norway	1947-2009	3.4	5.1	n.a.	n.a.	5.4
38	Japan	1946-2009	7.0	4.0	1.0	0.7	7.0
39	Italy	1951-2009	5.4	2.1	1.8	1.0	5.6
40	Ireland	1948-2009	4.4	4.5	4.0	2.4	2.9
41	Greece	1970-2009	4.0	0.3	2.7	2.9	13.3
42	Germany	1946-2009	3.9	0.9	n.a.	n.a.	3.2
43	France	1949-2009	4.9	2.7	3.0	n.a.	5.2
44	Finland	1946-2009	3.8	2.4	5.5	n.a.	7.0
45	Denmark	1950-2009	3.5	1.7	2.4	n.a.	5.6
46	Canada	1951-2009	1.9	3.6	4.1	n.a.	2.2
47	Belgium	1947-2009	n.a.	4.2	3.1	2.6	n.a.
48	Austria	1948-2009	5.2	3.3	-3.8	n.a.	5.7
49	Australia	1951-2009	3.2	4.9	4.0	n.a.	5.9
50							
51			4.1	2.8	2.8	=AVERAGE(L30:L44)	

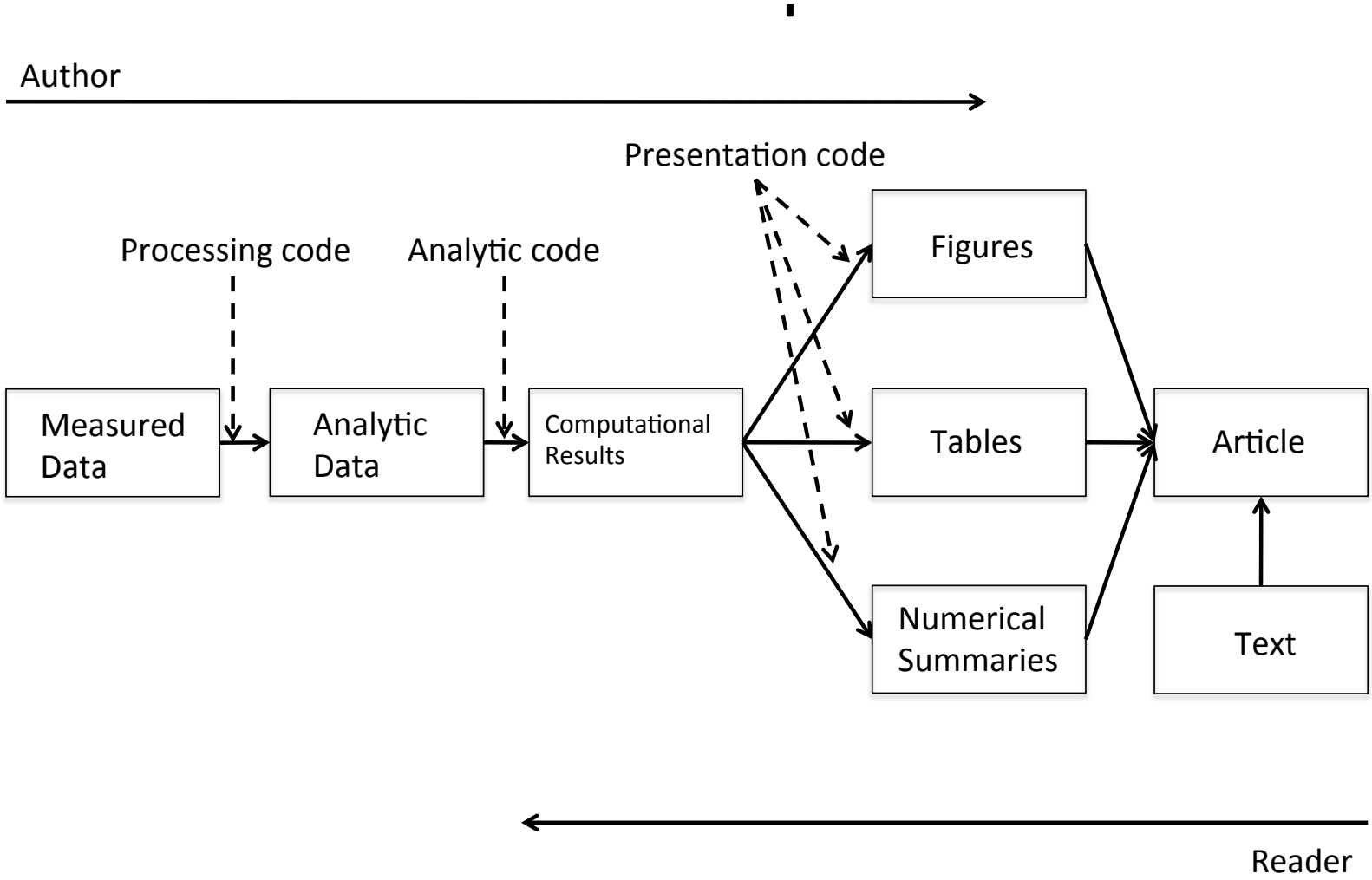
Replizierbarkeit

Der wissenschaftliche Standard verlangt als zentrales Kriterium für Forschung **Replizierbarkeit:** Ergebnisse müssen durch unabhängige

- Forscher
- Daten
- Analytische Methoden
- Laboratorien
- Instrumente

repliziert werden. Besonders wichtig ist dieses Kriterium für Forschung mit politischen, medizinischen, rechtlichen und wirtschaftlichen Konsequenzen. Leider kann dieses Ideal nicht immer erreicht werden (begrenzte Resourcen, einmalige Beobachtungen etc.)

Typische Forschungspipeline



Quelle: R.D.Peng, Johns Hopkins, 2013

Der “kleine Bruder” der Replizierbarkeit: Reproducible Research

Reproducible Research (reproduzierbare Forschung/Entwicklung): analytische Daten und Analysecode werden zur Verfügung gestellt, so dass andere die Ergebnisse reproduzieren können.

Es handelt sich also um eine Validierung nur der Datenanalyse, nicht der gesamten Studie.

Bestandteile:

- Analytische Daten
- Analysecode
- Dokumentation von Code und Daten
- Standardisierte Verteilungsmethoden

Reproduzierbarkeit und maschinelles Lernen

- Reproducible Research ist ein zentrales Konzept im maschinellen Lernen: die Synthese eines Algorithmus zur Lösung eines Problems aus Beispielen ist eine Form der Datenanalyse und damit ideal für Reproducible Research geeignet.
- Reproduzierbarkeit ist nicht nur in der Forschung wichtig, sondern genauso in der Industrie, um Ergebnisse effektiv und nachvollziehbar weitergeben zu können.
- Viele wissenschaftliche Fachzeitschriften (z.B. Science) verlangen mittlerweile die zusätzliche Veröffentlichung der analytischen Daten und des Analysecodes, v.a. in den "Computational X" (z.B. Computational Physics, Computational Neuroscience, …)
- Die veröffentlichten Datenbanken können zu "Megadatenbanken" zusammengefasst werden, wodurch neuartige Studien und Untersuchungsmethoden möglich werden.

Skripte statt Point & Click

- **Goldene Regel:** die gesamte Analyse wird geskriptet (auch wenn manche Analyseschritte nur einmal durchgeführt werden).
- Braucht Skriptsprache (z.B. Python, R, Matlab, Julia, …) und als Code ausführbare Software (keine GUIs).
- Der gesamte Code muss mithilfe eines Versionskontrollsystems (z.B. Git, Mercurial, SVN, …) verwaltet werden. Falls nicht zu groß, sollten dort auch die Daten und die zugehörige Dokumentation liegen.
- Eingesetzte Hardware, Betriebssystem, Versionsnummern der Software muss zusammen mit den produzierten Ergebnissen dokumentiert werden.
- Erzeugte Daten sollten so benannt werden, dass erkennbar wird, welches Skript sie erzeugt hat.
- Welche Daten mit welchem Skript bearbeitet wurden, sollte in einem README-File dokumentiert werden.

Umgang mit Daten

- Code und (nicht zu große) Dateien können auf einfache Weise über Repositorien wie z.B. Github - http://www.github.com oder Bitbucket - http://www.bitbucket.com zur Verfügung gestellt werden.
- Bei Daten von Websites muss immer zusätzlich die URL, Zeitpunkt des Downloads und eine passende Beschreibung (z.B. Variablennamen, physikalische Einheiten, Informationen zum experimentellen Aufbau) in einem README-File abgelegt werden (Format: Markdown, Text, evtl. Word).
- Nach Möglichkeit die Dateien über Kommandozeilenbefehle (z.B. git clone, wget) oder in Skripten herunterladen, keine Links in Browsern anklicken.
- Möglichst keine proprietären Dateiformate verwenden.
- Kein Editieren der Daten von Hand in Spreadsheets, keine Reorganisation oder Aufspaltung der Daten von Hand (falls doch: Vorgehensweise genau dokumentieren).

Hausaufgabe

Die in dieser Vorlesung eingesetzte Skriptsprache ist Python. Machen Sie sich, falls notwendig, bis zur nächsten Vorlesung mit den Grundstrukturen von Python vertraut.

Sinnvolle Einstiegsdokumente sind:

- Python in one easy lesson: http://cs.stanford.edu/people/nick/python-in-one-easy-lesson/
- Etwas ausführlicher: http://nbviewer.ipython.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-1-Introduction-to-Python-Programming.ipynb
- Offizielles Python-Tutorial: https://docs.python.org/2/tutorial/

Installation:

- Am einfachsten über die Distribution *Ananconda*: https://www.anaconda.com/